

Reti di Calcolatori T

Appello del 10/02/2021

Compito 1

Cognome:
Nome:
Matricola:

Tempo a disposizione: 3h

È obbligatorio inserire Cognome, Nome, e numero di Matricola all'inizio di ogni file sorgente, pena la non valutazione del compito, che viene stampato in modo automatico solo in caso siano presenti gli elementi detti sopra.

Si devono consegnare tutti i file sorgente e tutti gli eseguibili prodotti singolarmente (per favore, solo quelli relativi ai file sorgente consegnati!!).

La prova intende valutare le capacità progettuali e di programmazione sia in ambiente Java che in ambiente C, pertanto è consigliato sviluppare entrambe le soluzioni richieste al meglio.

Si tolga tutto il codice (e commenti) non rilevanti. Organizzate al meglio le strutture dati necessarie. In entrambi gli esercizi, sia in Java che in C, si effettuino gli opportuni controlli sui parametri della richiesta e si gestiscano le eccezioni, tenendo presente i criteri secondo cui si possa ripristinare il funzionamento del programma oppure si debba forzarne la terminazione.

Leggete bene le specifiche del problema prima di impegnarvi "a testa bassa" nello sviluppo delle singole parti. Naturalmente, ci aspettiamo che i componenti da consegnare siano stati provati e siano funzionanti.

Si richiede il progetto dei servizi **NoleggiaSci**, per la gestione del noleggio sci di un negozio di articoli sportivi. I servizi di NoleggiaSci mantengono, per ogni paio di sci, le seguenti informazioni: un **identificatore**, unico all'interno del sistema; il **modello** dello sci; per ogni sci noleggiato, la **data**, **giorno**, **mese** e **anno** di inizio noleggio (per semplicità si ipotizzano tutti mesi da 30 giorni) e il **numero di giorni** di noleggio; il **costo giornaliero di noleggio** (la durata minima del noleggio è un giorno); il **nome di un file .jpg** (del direttorio corrente del server) con una foto dello sci.

Si vogliono realizzare le funzionalità di gestione:

1. **elimina un paio di sci**: questa operazione richiede l'identificatore e elimina lo sci, modificando la struttura dati ed eliminando il relativo file, e restituisce l'esito dell'operazione;
2. **richiesta di noleggio di un paio di sci**: questa operazione richiede l'identificatore, il giorno, mese e anno di inizio noleggio e la durata in giorni del noleggio, quindi aggiorna la struttura dati e restituisce l'esito dell'operazione;
3. **download degli sci noleggiabili di un certo modello**: questa operazione richiede il modello, quindi restituisce la lista degli sci noleggiabili per tale modello e i relativi file immagine;
4. **valutazione costo di un noleggio**: questa operazione richiede l'identificatore e restituisce il costo complessivo del noleggio per lo sci richiesto (per tutta la durata del noleggio).

Si progetti con particolare attenzione la **struttura dati** che mantiene lo stato, fino ad un massimo di K sci (L e -1 – o anche -1/-1/-1 – per libero o non presente a default), da implementare e inizializzare nei diversi ambienti richiesti, Java e C.

Identificatore	Data	Giorni	Modello	Costo giornaliero	Nome file foto
L	-1/-1/-1	-1	-1	-1	L
X12AB	12/12/2012	14	Volki Shark	7	VolkiShark1.jpg
ACD14	-1/-1/-1	-1	Fisher Snow	5	FisherSnow.jpg
Y23CC	23/12/2012	7	Volki Shark	7	VolkiShark2.jpg
...
L	-1/-1/-1	-1	-1	-1	-1

Si considerino e si segnalino le possibilità di interferenze fra le operazioni, evitandole dove necessario.

Parte Java

Utilizzando **java RMI** sviluppare un'applicazione C/S che consenta di effettuare le operazioni remote per:

- eliminare un paio di sci;
- richiedere il noleggio di un paio di sci.

Il progetto RMI si basa su un'interfaccia (contenuta nel file *RMI_interfaceFile.java*) in cui vengono definiti i metodi invocabili in remoto dal client:

- Il metodo **elimina_sci** accetta come parametro d'ingresso l'identificatore, elimina lo sci nella struttura dati, eliminando anche il relativo file sul server, e restituisce l'esito dell'operazione, positivo se la modifica è andata a buon fine, negativo altrimenti, ad esempio se non esiste uno sci con l'identificatore indicato.
- Il metodo **noleggia_sci** accetta come parametri d'ingresso l'identificatore, il giorno, mese e anno di inizio noleggio e la durata in giorni del noleggio, aggiorna la struttura dati e restituisce l'esito dell'operazione richiesta, positivo se l'operazione è andata a buon fine, negativo altrimenti, ad esempio se gli sci non sono presenti nella struttura dati o se sono già stati prenotati per una qualunque durata.

Si progettino le classi:

- **RMI_Server** (contenuta nel file *RMI_Server.java*), che implementa i metodi del server invocabili in remoto;
- **RMI_Client** (contenuta nel file *RMI_Client.java*), il processo che realizza l'interazione con l'utente **proponendo ciclicamente i servizi** che utilizzano i due metodi remoti, e stampa a video i risultati, fino alla fine del file di input da tastiera.

Parte C

Progettare un **servitore multiservizio (usando obbligatoriamente una select)** che consenta di effettuare le operazioni remote per:

- download degli sci noleggiabili di un certo modello;
- valutare il costo di un noleggio.

Più in dettaglio:

- Il **client_stream** è organizzato come un **processo filtro ciclico fino alla fine del file di input** e realizza la funzionalità di **download degli sci noleggiabili di un certo modello** utilizzando **socket stream** e un'unica connessione per la intera sessione.

Per ogni operazione, il client richiede all'utente e invia al server il modello, quindi riceve la lista di sci corrispondenti stampandoli a video e ricevendo e salvando, per ciascuno di essi, il relativo file immagine sul file system locale del client (nel direttorio corrente).

- Il **client_datagram** è organizzato come un **processo filtro ciclico fino alla fine del file di input** e realizza la funzionalità di **valutazione del costo di un noleggio** utilizzando **socket datagram**.

Per ogni richiesta, il client richiede all'utente e invia al server l'identificatore; quindi riceve l'esito dell'operazione e lo stampa a video.

- Il **server** principale unico discrimina le richieste utilizzando la primitiva select: il **server gestisce in modo parallelo** la funzionalità di download degli sci noleggiabili di un certo modello; mentre la funzionalità di valutazione del costo di un noleggio **può essere realizzata in modo o seriale o parallelo**: si motivi la scelta fatta.

Per ogni richiesta di **download degli sci noleggiabili di un certo modello**, il figlio riceve il modello e trasmette al client la intera lista informazione degli sci corrispondenti a tale modello e non già noleggiati, e inviando, per ciascuno di essi, il relativo file immagine.

Per ogni richiesta di **valutazione del costo di un noleggio**, il server riceve l'identificatore e restituisce al client il costo totale del noleggio per tutte le giornate: il costo totale del noleggio nel caso in cui lo sci venga identificato e sia effettivamente noleggiato, oppure un esito negativo, ad esempio se lo sci non è stato noleggiato oppure non è presente nella struttura dati.